Evaluating the visualization of what a Deep Neural Network has learned

Wojciech Samek† *Member, IEEE,* Alexander Binder†, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller, *Member, IEEE,*

***Abstract*—Deep Neural Networks (DNNs) have demonstrated impressive performance in complex machine learning tasks such as image classification or speech recognition. However, due to their multi-layer nonlinear structure, they are not transparent, i.e., it is hard to grasp *what* makes them arrive at a particular classification or recognition decision given a new unseen data sample. Recently, several approaches have been proposed enabling one to understand and interpret the reasoning embodied in a DNN for a single test image. These methods quantify the "importance" of individual pixels wrt the classification decision and allow a visualization in terms of a heatmap in pixel/input space. While the usefulness of heatmaps can be judged subjectively by a human, an objective quality measure is missing. In this paper we present a general methodology based on region perturbation for evaluating ordered collections of pixels such as heatmaps. We compare heatmaps computed by three different methods on the SUN397, ILSVRC2012 and MIT Places data sets. Our main result is that the recently proposed Layer-wise Relevance Propagation (LRP) algorithm qualitatively and quantitatively provides a better explanation of what made a DNN arrive at a particular classification decision than the sensitivity-based approach or the deconvolution method. We provide theoretical arguments to explain this result and discuss its practical implications. Finally, we investigate the use of heatmaps for unsupervised assessment of neural network performance.**

***Index Terms*—Convolutional Neural Networks, Explanation, Heatmapping, Relevance Models, Image Classification.**

I. Introduction

Deep Neural Networks (DNNs) are powerful methods for solving large scale real world problems such as automated image classification [1], [2], [3], [4], natural language processing [5], [6], human action recognition [7], [8], or physics [9]; see also [10]. Since DNN training methodologies (unsupervised pretraining, dropout, parallelization, GPUs etc.) have been improved [11], DNNs are recently able to harvest extremely large amounts of training data and can thus achieve record performances in many research fields. At the same time, DNNs are generally conceived as black box methods, and users might consider this lack of transparency a drawback in practice. Namely, it is difficult to intuitively and quantitatively understand the result of DNN inference, i.e., for an *individual* novel input data point, *what* made the trained DNN model arrive at a particular response. Note that this aspect differs from feature selection [12], where the question is: which features are on average salient for the *ensemble* of training data.

Only recently, the transparency problem has been receiving more attention for general nonlinear estimators [13], [14], [15], [16]. Several methods have been developed to understand what a DNN has learned [17], [18]. While in DNN a large body of work is dedicated to visualize particular neurons or neuron layers [1], [19], [20], [21], [22], [23], [24], we focus here on methods which visualize the impact of particular regions of a given and fixed single image for a prediction of this image. Zeiler and Fergus [19] have proposed in their work a network propagation technique to identify patterns in a given input image that are linked to a particular DNN prediction. This method runs a backward algorithm that reuses the weights at each layer to propagate the prediction from the output down to the input layer, leading to the creation of meaningful patterns in input space. This approach was designed for a particular type of neural network, namely convolutional nets with max-pooling and rectified linear units. A limitation of the deconvolution method is the absence of a particular theoretical criterion that would directly connect the predicted output to the produced pattern in a quantifiable way. Furthermore, the usage of image-specific information for generating the backprojections in this method is limited to max-pooling layers alone. Further previous work has focused on understanding non-linear learning methods such as DNNs or kernel methods [14], [25], [26] essentially by sensitivity analysis in the sense of scores based on partial derivatives at the given sample. Partial derivatives look at local sensitivities detached from the decision boundary of the classifier. Simonyan et al. [26] applied partial derivatives for visualizing input sensitivities in images classified by a deep neural network. Note that although [26] describes a Taylor series, it relies on partial derivatives at the given image for computation of results. In a strict sense partial derivatives do not explain a classifier's decision

This work was supported by the Brain Korea 21 Plus Program through the National Research Foundation of Korea funded by the Ministry of Education. This work was also supported by the grant DFG (MU 987/17-1) and by the German Ministry for Education and Research as Berlin Big Data Center BBDC (01IS14013A). This publication only reflects the authors views. Funding agencies are not liable for any use that may be made of the information contained herein. *Asterisks indicate corresponding author.*

*W. Samek is with Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany. (e-mail: wojciech.samek@hhi.fraunhofer.de)

*A. Binder is with the ISTD Pillar, Singapore University of Technology and Design (SUTD), Singapore, and with the Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany. (e-mail: alexander_binder@sutd.edu.sg)

G. Montavon is with the Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany. (e-mail: gregoire.montavon@tu-berlin.de)

S. Lapuschkin is with Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany. (e-mail: sebastian.lapuschkin@hhi.fraunhofer.de)

*K.-R. Müller is with the Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany, and also with the Department of Brain and Cognitive Engineering, Korea University, Seoul 136-713, Korea (e-mail: klaus-robert.mueller@tu-berlin.de)

† WS and AB contributed equally

(*"what speaks for the presence of a car in the image"*), but rather tell us *what change would make the image more or less belong to the category car*. As shown later these two types of explanations lead to very different results in practice. An approach, Layer-wise Relevance Propagation (LRP), which is applicable to arbitrary types of neural unit activities (even if they are non-continuous) and to general DNN architectures (and Fisher vectors classifiers [27]) has been proposed by Bach et al. [28]. This work aims at explaining the difference of a prediction $f(x)$ relative to the neutral state $f(x) = 0$. The LRP method relies on a conservation principle to propagate the prediction back without using gradients. This principle ensures that the network output activity is fully redistributed through the layers of a DNN onto the input variables, i.e., neither positive nor negative evidence is lost.

In the following we will denote the visualizations produced by the above methods as heatmaps. While per se a heatmap is an interesting and intuitive tool that can already allow to achieve transparency, it is difficult to quantitatively evaluate the quality of a heatmap. In other words we may ask: what exactly makes a "good" heatmap. A human may be able to intuitively assess the quality of a heatmap, e.g., by matching with a prior of what is regarded as being relevant (see Figure 1). For practical applications, however, an automated objective and quantitative measure for assessing heatmap quality becomes necessary. Note that the validation of heatmap quality is important if we want to use it as input for further analysis. For example we could run computationally more expensive algorithms only on relevant regions in the image, where relevance is detected by a heatmap.

In this paper we contribute by

- pointing to the issue of how to objectively evaluate the quality of heatmaps. To the best of our knowledge this question has not been raised so far.
- introducing a generic framework for evaluating heatmaps which extends the approach in [28] from binary inputs to color images.
- comparing three different heatmap computation methods on three large data-sets and noting that the relevance-based LRP algorithm [28] is more suitable for explaining the classification decisions of DNNs than the sensitivity-based approach [26] and the deconvolution method [19].
- investigating the use of heatmaps for assessment of neural network performance.

The next section briefly introduces three existing methods for computing heatmaps. Section III discusses the heatmap evaluation problem and presents a generic framework for this task. Two experimental results are presented in Section IV: The first experiment compares different heatmapping algorithms on SUN397 [29], ILSVRC2012 [30] and MIT Places [31] data sets and the second experiment investigates the correlation between heatmap quality and neural network performance on the CIFAR-10 data set [32]. We conclude the paper in Section V and give an outlook.

II. Understanding DNN Prediction

In the following we focus on images, but the presented techniques are applicable to any type of input domain whose elements can be processed by a neural network.

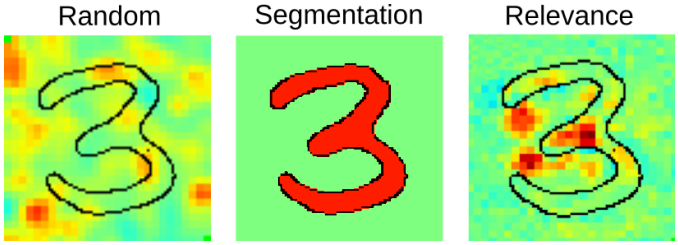

Fig. 1. Comparison of three exemplary heatmaps for the image of a '3'. Left: The randomly generated heatmap lacks interpretable information. Middle: The segmentation heatmap (binary values) focuses on the whole digit without indicating what parts of the image were particularly relevant for classification. Since it does not suffice to consider only the highlighted pixels for distinguishing an image of a '3' from images of an '8' or a '9', this heatmap is not useful for explaining classification decisions. Right: A relevance heatmap indicates which parts of the image are used by the classifier. Here the heatmap reflects human intuition very well because the horizontal bar together with the missing stroke on the left are strong evidence that the image depicts a '3' and not any other digit.

Let us consider an image $\boldsymbol{x} \in \mathbb{R}^d$, decomposable as a set of pixel values $\boldsymbol{x} = \{x_p\}$ where p denotes a particular pixel, and a classification function $f : \mathbb{R}^d \to \mathbb{R}^+$. The function value $f(\boldsymbol{x})$ can be interpreted as a score indicating the certainty of the presence of a certain type of object(s) in the image. Such functions can be learned very well by a deep neural network. Throughout the paper we assume neural networks to consist of multiple layers of neurons, where neurons are activated as

$$a_j^{(l+1)} = \sigma\Big(\sum_i z_{ij} + b_j^{(l+1)}\Big) \tag{1}$$

$$\text{with} \quad z_{ij} = a_i^{(l)} w_{ij}^{(l,l+1)} \tag{2}$$

The sum operator runs over all lower-layer neurons that are connected to neuron j, where $a_i^{(l)}$ is the activation of a neuron i in the previous layer, and where z_{ij} is the contribution of neuron i at layer l to the activation of the neuron j at layer $l+1$. The function σ is a nonlinear monotonously increasing activation function, $w_{ij}^{(l,l+1)}$ is the weight and $b_j^{(l+1)}$ is the bias term.

A *heatmap* $\boldsymbol{h} = \{h_p\}$ assigns each pixel p a value $h_p = \mathcal{H}(\boldsymbol{x}, f, p)$ according to some function $\mathcal{H}$, typically derived from a class discriminant f. Since $\boldsymbol{h}$ has the same dimensionality as $\boldsymbol{x}$, it can be visualized as an image. In the following we review three recent methods for computing heatmaps, all of them performing a backward propagation pass on the network: (1) a sensitivity analysis based on neural network partial derivatives, (2) the so-called deconvolution method and (3) the layer-wise relevance propagation algorithm. Figure 2 briefly summarizes the methods.

A. Sensitivity Heatmaps

A well-known tool for interpreting non-linear classifiers is sensitivity analysis [14]. It was used by Simonyan et al. [26] to compute saliency maps of images classified by neural networks. In this approach the sensitivity of a pixel h_p is computed by using the norm $\|\cdot\|_{\ell_q}$ over partial derivatives

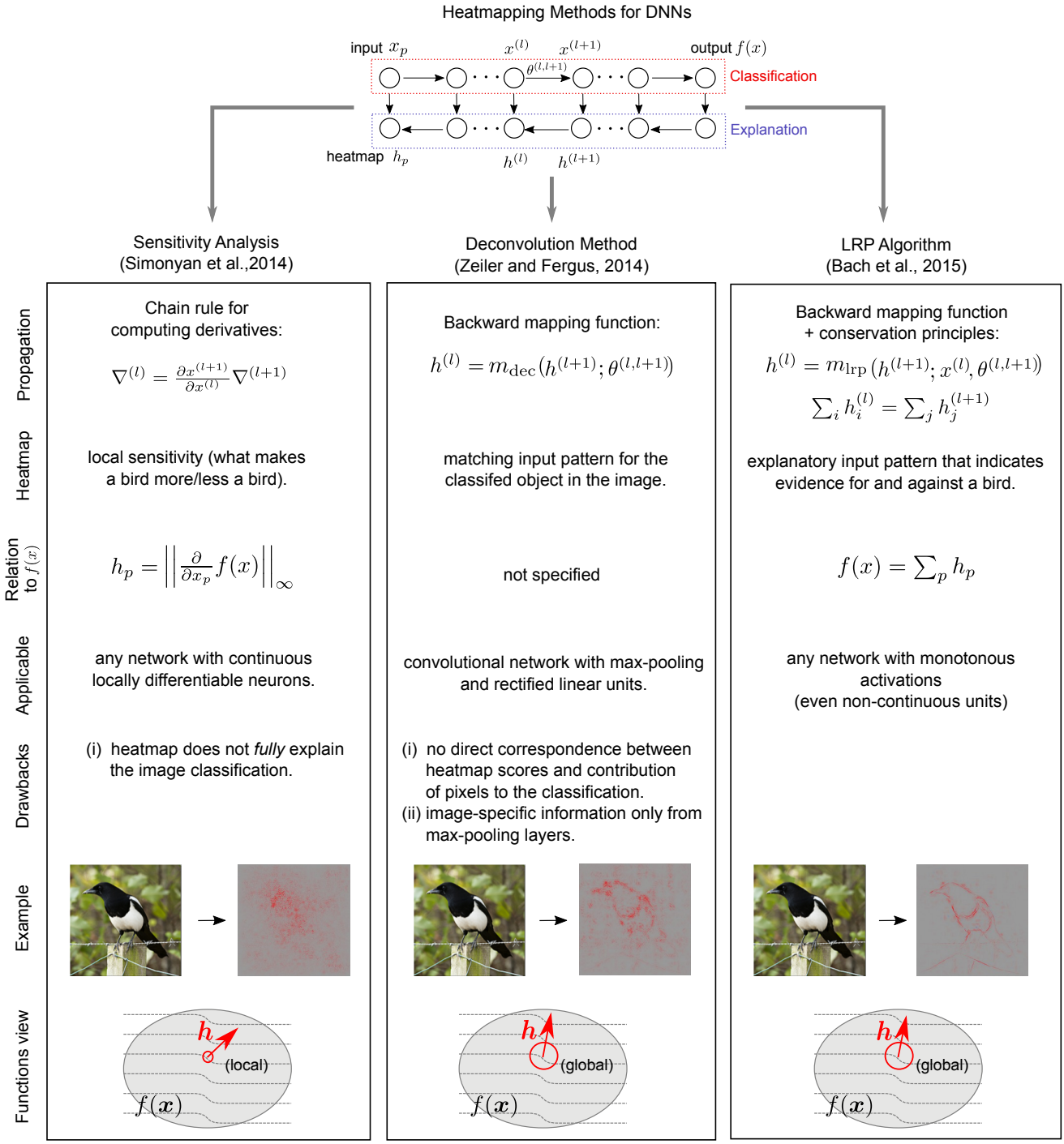

Fig. 2. Comparison of the three heatmap computation methods used in this paper. Left: Sensitivity heatmaps are based on partial derivatives, i.e., measure which pixels, when changed, would make the image belong less or more to a category (local explanations). The method is applicable to generic architectures with differentiable units. Middle: The deconvolution method applies a convolutional network g to the output of another convolutional network f. Network g is constructed in a way to "undo" the operations performed by f. Since negative evidence is discarded and scores are not normalized during the backpropagation, the relation between heatmap scores and the classification output $f(x)$ is unclear. Right: Layer-wise Relevance Propagation (LRP) exactly decomposes the classification output $f(x)$ into pixel relevances by observing the layer-wise conservation principle, i.e., evidence for or against a category is not lost. The algorithm does not use gradients and is therefore applicable to generic architectures (including nets with non-continuous units). LRP globally explains the classification decision and heatmap scores have a clear interpretation as evidence for or against a category.

([26] used $q = \infty$) for the color channel c of a pixel p:

$$h_p = \left|\left| \left(\frac{\partial}{\partial x_{p,c}} f(\boldsymbol{x}) \right)_{c \in (r,g,b)} \right|\right|_{\ell_q} \quad (3)$$

This quantity measures how much small changes in the pixel value locally affect the network output. Large values of h_p denote pixels which largely affect the classification function f if changed. Note that the direction of change (i.e., sign of the partial derivative) is lost when using the norm. Partial

derivatives are obtained efficiently by running the backpropagation algorithm [33] throughout the multiple layers of the network. The backpropagation rule from one layer to another layer, where $x^{(l)}$ and $x^{(l+1)}$ denote the neuron activities at two consecutive layers is given by:

$$\frac{\partial f}{\partial x^{(l)}} = \frac{\partial x^{(l+1)}}{\partial x^{(l)}} \frac{\partial f}{\partial x^{(l+1)}} \tag{4}$$

The backpropagation algorithm performs the following operations in the various layers:

Unpooling: The gradient signal is redirected onto the input neuron(s) to which the corresponding output neuron is sensitive. In the case of max-pooling, the input neuron in question is the one with maximum activation value.

Nonlinearity: Denoting by $z_i^{(l)}$ the preactivation of the ith neuron of the lth layer, backpropagating the signal through a rectified linear unit (ReLU) defined by the map $z_i^{(l)} \rightarrow \max(0, z_i^{(l)})$ corresponds to multiplying the backpropagated gradient signal by the step function $1_{\{z_i^{(l)}>0\}}$. The multiplication of the signal by the step function makes the backward mapping discontinuous, and consequently strongly local.

Filtering: The gradient signal is convolved by a transposed version of the convolutional filter used in the forward pass.

In the experiments, we compute heatmaps by using Eq. 3 with the norms $q = \{2, \infty\}$.

B. Deconvolution Heatmaps

Another method for heatmap computation was proposed in [19] and uses a process termed deconvolution. Similarly to the backpropagation method to compute the function's gradient, the idea of the deconvolution approach is to map the activations from the network's output back to pixel space using a backpropagation rule

$$R^{(l)} = m_{\text{dec}}(R^{(l+1)}; \theta^{(l,l+1)}). \tag{5}$$

Here, $R^{(l)}, R^{(l+1)}$ denote the backward signal as it is backpropagated from one layer to the previous layer, m_{dec} is a predefined function that may be different for each layer and $\theta^{(l,l+1)}$ is the set of parameters connecting two layers of neurons. This method was designed for a convolutional net with max-pooling and rectified linear units, but it could also be adapted in principle for other types of architectures. The following set of rules is applied to compute deconvolution heatmaps.

Unpooling: The locations of the maxima within each pooling region are recorded and these recordings are used to place the relevance signal from the layer above into the appropriate locations. For deconvolution this seems to be the only place besides the classifier output where image information from the forward pass is used, in order to arrive at an image-specific explanation.

Nonlinearity: The relevance signal at a ReLU layer is passed through a ReLU function during the deconvolution process.

Filtering: In a convolution layer, the transposed versions of the trained filters are used to backpropagate the relevance signal. This projection does not depend on the neuron activations $x^{(l)}$.

The unpooling and filtering rules are the same as those derived from gradient propagation (i.e., those used in Section II-A). The propagation rule for the ReLU nonlinearity differs from backpropagation: Here, the backpropagated signal is not multiplied by a discontinuous step function, but is instead passed through a rectification function similar to the one used in the forward pass. Note that unlike the indicator function, the rectification function is continuous. For deconvolution we apply the same color channel pooling methods (2-norm, ∞-norm) as for sensitivity analysis.

C. Relevance Heatmaps

Layer-wise Relevance Propagation (LRP) [28] is a principled approach to decompose a classification decision into pixel-wise *relevances* indicating the contributions of a pixel to the overall classification score. The approach is derived from a layer-wise conservation principle [28], which forces the propagated quantity (e.g., evidence for a predicted class) to be preserved between neurons of two adjacent layers. Denoting by $R_i^{(l)}$ the relevance associated to the ith neuron of layer l and by $R_j^{(l+1)}$ the relevance associated to the jth neuron in the next layer, the conservation principle requires that

$$\textstyle\sum_i R_i^{(l)} = \sum_j R_j^{(l+1)} \tag{6}$$

where the sums run over all neurons of the respective layers. Applying this rule repeatedly for all layers, the heatmap resulting from LRP satisfies $\sum_p h_p = f(\boldsymbol{x})$ where $h_p = R_p^{(1)}$ and is said to be consistent with the evidence for the predicted class. Stricter definitions of conservation that involve only subsets of neurons can further impose that relevance is locally redistributed in the lower layers. The propagation rules for each type of layers are given below:

Unpooling: Like for the previous approaches, the backward signal is redirected proportionally onto the location for which the activation was recorded in the forward pass.

Nonlinearity: The backward signal is simply propagated onto the lower layer, ignoring the rectification operation. Note that this propagation rule satisfies Equation 6.

Filtering: Bach et al. [28] proposed two relevance propagation rules for this layer, that satisfy Equation 6. Let $z_{ij} = a_i^{(l)} w_{ij}^{(l,l+1)}$ be the weighted activation of neuron i onto neuron j in the next layer. The first rule is given by:

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j} + \epsilon \,\text{sign}(\sum_{i'} z_{i'j})} R_j^{(l+1)} \tag{7}$$

The intuition behind this rule is that lower-layer neurons that mostly contribute to the activation of the higher-layer neuron receive a larger share of the relevance R_j of the neuron j. The neuron i then collects the relevance associated to its contribution from all upper-layer neurons j. A downside of this propagation rule (at least if $\epsilon = 0$) is that the denominator may tend to zero if lower-level contributions to neuron j cancel each other out. The numerical instability can be overcome by

setting $\epsilon > 0$. However in that case, the conservation idea is relaxated in order to gain better numerical properties. A way to achieve exact conservation is by separating the positive and negative activations in the relevance propagation formula, which yields the second formula:

$$R_i^{(l)} = \sum_j \left(\alpha \cdot \frac{z_{ij}^+}{\sum_{i'} z_{i'j}^+} + \beta \cdot \frac{z_{ij}^-}{\sum_{i'} z_{i'j}^-} \right) R_j^{(l+1)}. \quad (8)$$

Here, z_{ij}^+ and z_{ij}^- denote the positive and negative part of z_{ij} respectively, such that $z_{ij}^+ + z_{ij}^- = z_{ij}$. We enforce $\alpha + \beta = 1$ in order for the relevance propagation equations to be conservative layer-wise. It should be emphasized that unlike gradient-based techniques, the LRP formula is applicable to non-differentiable neuron activation functions. The LRP algorithm does not multiply its backward signal by a discontinuous function. Therefore, relevance heatmaps also favor the emergence of global features, that allow for full explanation of the class to be predicted.

In the experiments section we use for consistency the same settings as in [28] without having optimized the parameters, namely the LRP variant from Equation (8) with $\alpha = 2$ and $\beta = -1$ (which will be denoted as LRP in subsequent figures), and twice LRP from Equation (7) with $\epsilon = 0.01$ and $\epsilon = 100$. An implementation of LRP is described in [34] and can be downloaded from `http://heatmapping.org`.

D. Theoretical Analysis

In the following we investigate the advantages and limitations of the presented heatmapping methods. We show that LRP has six desirable properties, which are not or only partly satisfied by sensitivity analysis and the deconvolution method.

1) Global Explanations: A desired property of heatmapping methods is that they provide global explanations, e.g., indicate what are the features that compose a given car. This property is satisfied by LRP and to some extent by the deconvolution method, but not by sensitivity analysis. The latter gives for every pixel a direction in RGB-space in which the prediction increases or decreases, but it does not indicate directly whether a particular region contains evidence for or against the prediction made by a classifier. Thus, it provides *local* explanations, e.g., indicate what makes a given car look more/less like a car. The subplot a) of Figure 3 shows the qualitative difference between gradient- and LRP-type explanations. In terms of gradients it is a valid explanation to put high norms on the empty street, because there exists a direction in the input space in which the classifier prediction can be increased by putting motor-bike like structures in there. However, from a global explanation perspective the streets are not very indicative of the class scooter in this particular image. This example shows that regions consisting of pure background may have a notable sensitivity, which makes gradient-type explanations noisier than LRP and deconvolution heatmaps (see also Figure 6).

2) Continuous Explanations: Another desired property of heatmapping methods is that they provide continuous explanations, i.e., small variations in the input should not result in large changes in the heatmap. Also this property is not satisfied by gradient-type methods. The multiplication of the signal by an indicator function in the rectification layer (see Section II-A) makes the backward mapping of gradient-type methods discontinuous, i.e., they may abruptly switch from considering one feature as being highly relevant to considering another feature as being the most important one. The subplot b) of Figure 3 shows that sensitivity analysis of a two-dimensional function $f(x_1, x_2)$ results in discontinuous explanations (red arrows abruptly change direction), whereas LRP (and also deconvolution) does not show this behavior and thus provides more reliable explanations.

3) Image Specific Explanations: Salient features represent average explanations of what distinguishes one image category from another. For individual images these explanations may be meaningless or even wrong. The deconvolution method only implicitly takes into account properties of individual images through the unpooling operation. The backprojection over filtering layers is independent of the individual image. Thus, when applied to neural networks without a pooling layer, both methods will not provide individual (image specific) explanations, but rather average salient features. LRP's rule for filtering layers on the other hand takes into account *both* filter weights and lower-layer neuron activations. This allows for individual explanations even in a neural network without pooling layers. The subplot c) of Figure 3 demonstrates this property on a simple example. We compare the explanations provided by the deconvolution method and LRP for a neural network without pooling layers trained on the MNIST data set. One can see that LRP provides individual explanations for all images in the sense that when the digit in the image is slightly rotated, then the heatmap adapts to this rotation and highlights the relevant regions of this particular rotated digit. The deconvolution heatmap on the other hand is not image-specific because it only depends on the weights and not the neuron activations. If pooling layers were present in the network, then the deconvolution approach would implicitly adapt to the specific image through the unpooling operation. Still we consider this information important to be included when backprojecting over filtering layers, because neurons with large activations for a specific image should be regarded as more relevant, thus should backproject a larger share of the relevance.

4) Positive and Negative Evidence: In contrast to sensitivity analysis and the deconvolution method, LRP provides signed explanations and thus distinguishes between positive evidence, supporting the classification decision, and negative evidence, speaking against the prediction. The subplot d) of Figure 3 shows that LRP responses can be well interpreted in that way; the red and blue color represent positive and negative evidence, respectively. In particular, when backpropagating the (artificial) classification decision that the image has been classified as '9', LRP provides a very intuitive explanation, namely that in the left upper part of the image the missing stroke closing the loop (blue color) speaks against the fact that this is a '9' whereas the missing stroke in the left lower part of the image (red color) supports this decision. The deconvolution method (and the gradient-type explanation) does not allow such interpretation.

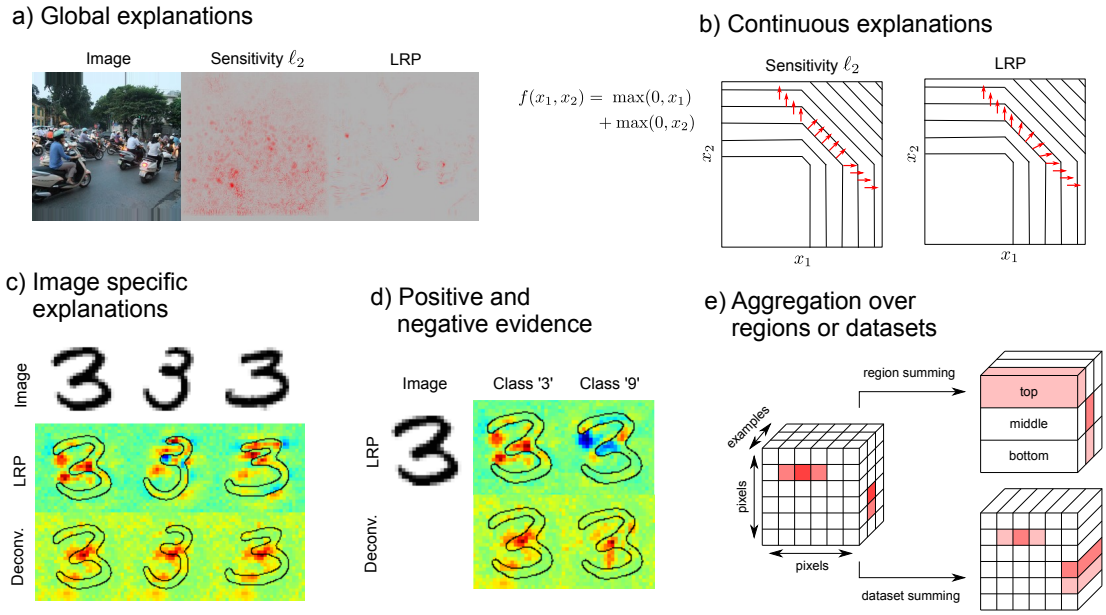

Fig. 3. Desirable properties of heatmapping methods. a) LRP provides global explanations, i.e., indicates what are the features that explain the prediction "scooter", whereas sensitivity analysis provides local explanations, i.e., shows what would make the image less or more belong to the category "scooter". b) LRP provides continuous explanations (red arrows) which change slowly with the input, whereas the explanations provided by sensitive analysis change abruptly due to discontinuities. c) LRP provides image specific explanations, because it takes into account the weights and the activations, whereas the deconvolution method only considers the weights and thus produces the same heatmaps for different input images. d) LRP distinguishes between positive evidence supporting a prediction (red) and negative evidence speaking against it (blue), whereas deconvolution does not provide signed explanations. e) The conservation property of LRP provided a meaningful normalization of the heatmaps, thus allows one to aggregate the explanations over regions or datasets.

5) Aggregating over Regions or Datasets: Aggregating explanations over image regions or over different datasets (see subplot e) of Figure 3) is a desired property requiring meaningful normalization of the pixel-wise scores. The explanations computed by sensitivity analysis and the deconvolution method are not normalized, so that aggregation may lead to meaningless results (e.g., when the heatmap of one image has values which are an order of magnitude larger than values of other heatmaps). The LRP scores on the other hand are directly related to the classification output through the conservation principle, thus are meaningfully normalized. This allows for a meaningful aggregation.

6) Relation to Classification Output: Another desirable property of heatmapping methods is an explicit mathematical relation between heatmap and the classification output, because this allows for an interpretation of the obtained scores. Such explicit relation exists for the gradient-type approach and for the LRP method (see formula in Figure 2). For the deconvolution method these relationships cannot be expressed analytically, because negative evidence ($R^{(l+1)} < 0$) is discarded during the backpropagation due to the application of the ReLU function and the backward signal is not normalized layer-wise, so that few dominant $R^{(l)}$ may largely determine the final heatmap scores.

Finally, an additional justification for the way LRP technically operates can be found in [35] where the method is shown for certain choices of parameters to perform a "deep Taylor decomposition" of the neural network function.

III. Evaluating Heatmaps

In this section, we introduce a set of methods to evaluate empirically the quality of a heatmapping technique. Although humans are able to intuitively assess the quality of a heatmap by matching with prior knowledge and experience of what is regarded as being relevant, defining objective criteria for heatmap quality is very difficult. In this paper we refrain from mimicking the complex human heatmap evaluation process which includes attention, interest point models and perception models of saliency [36], [37], [38], [39] for the reason that we are interested in finding those regions that are relevant for a given classifier.

Rather than representing human reasoning, or matching some ground truth on what is important in an image, heatmaps should reflect the machine learning classifier's own "view" on the classification problem, more precisely, identify the pixels used by the classifier to support its decision. Thus, heatmap quality does not only depend on the algorithms used to compute a heatmap, but also on the performance of the classifier, whose efficiency largely depends on the model being used, and the amount and quality of available training data. For example, if the training data does not contain images of the digits '3', then the classifier can not know that the absence of strokes in the left part of the image (see example in Figure 1) is important for distinguishing the digit '3' from digits '8' and '9'. Thus, explanations can only be as good as the data provided to the classifier. A random classifier will lead to uninformative heatmaps.

Note also that a heatmap differs from a segmentation mask (see Figure 1) in several ways: First, a heatmap can rightfully associate relevance to pixels outside the object to detect, for example, when the context of the object (e.g., water texture behind a boat) provides some useful information for classification. Second, segmentation masks are binary (i.e.,

they essentially determine whether a pixel is part or not of the object to detect). A heatmap, on the other hand, provides a gradation of pixel scores, which correspond to the degree of importance of each pixel for determining predicted class membership. Points of interest (e.g., eyes, or small objects), might concentrate as much information for determining the class as larger surfaces. A heatmap can also associate negative values to pixels that contradict the prediction.

A. Heatmap Evaluation Framework

More formally, a heatmap is an array of pixel-wise scores $(R_p)_p$, that indicates which pixels are *relevant* for a classification decision (i.e., which pixels make $f(\boldsymbol{x})$ large). A heatmap can be viewed as defining a subspace composed of pixels with high relevance scores, on which the function $f(\boldsymbol{x})$ must be scrutinized. We expect the pixels to which most relevance is associated to be those that are the most likely to destroy the value $f(\boldsymbol{x})$ if they are perturbed, in other words, we would like to test how fast the function value drops when moving in this subspace.

To test this expected behavior, we consider a greedy iterative procedure that consists of measuring how the class encoded in the image (e.g., as measured by the function f) disappears when we progressively remove information from the image $\boldsymbol{x}$, a process referred to as *region perturbation*, at the specified locations. The method is a generalization of the approach presented in [28], where the perturbation process is a state flip of the associated binary pixel values (single pixel perturbation). The method that we propose here applies more generally to *any* set of locations (e.g., local windows) and any local perturbation process such as local randomization or blurring.

We define a heatmap as an ordered set of locations in the image, where these locations might lie on a predefined grid.

$$\mathcal{O} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_L) \tag{9}$$

Each location $\boldsymbol{r}_p$ is for example a two-dimensional vector encoding the horizontal and vertical position on a grid of pixels. The ordering can either be chosen at hand, or be induced by a heatmapping function $h_p = \mathcal{H}(\boldsymbol{x}, f, \boldsymbol{r}_p)$, typically derived from a class discriminant f (see methods in Section II). The scores $\{h_p\}$ indicate how important the given location $\boldsymbol{r}_p$ of the image is for representing the image class. The ordering induced by the heatmapping function is such that for all indices of the ordered sequence $\mathcal{O}$, the following property holds:

$$(i < j) \Leftrightarrow \big(\mathcal{H}(\boldsymbol{x}, f, \boldsymbol{r}_i) > \mathcal{H}(\boldsymbol{x}, f, \boldsymbol{r}_j)\big) \tag{10}$$

Thus, locations in the image that are most relevant for the class encoded by the classifier function f will be found at the beginning of the sequence $\mathcal{O}$. Conversely, regions of the image that are mostly irrelevant will be positioned at the end of the sequence.

We consider a region perturbation process that follows the ordered sequence of locations. We call this process *most relevant first*, abbreviated as MoRF. The recursive formula is:

$$\begin{aligned} \boldsymbol{x}^{(0)}_{\text{MoRF}} &= \boldsymbol{x} \\ \forall\ 1 \leq k \leq L:\ \boldsymbol{x}^{(k)}_{\text{MoRF}} &= g(\boldsymbol{x}^{(k-1)}_{\text{MoRF}}, \boldsymbol{r}_k) \end{aligned} \tag{11}$$

where the function g removes information of the image $\boldsymbol{x}^{(k-1)}_{\text{MoRF}}$ at a specified location $\boldsymbol{r}_k$ (i.e., a single pixel or a local neighborhood) in the image. Throughout the paper we use a function g which replaces all pixels in a $m \times m$ neighborhood around $\boldsymbol{r}_k$ by randomly sampled (from uniform distribution) values. The choice of uniform distribution follows our intention to use a model-free method to generate perturbations. A model-based estimation of probabilities for patches may result in biases due to the model assumptions. Using the uniform distribution ensures that we treat all regions equally and explore the whole imaging space. Furthermore it ensures that we evaluate the behaviour of the classifier under perturbations that are off the data-manifold.

When comparing different heatmaps using a fixed $g(\boldsymbol{x}, \boldsymbol{r}_k)$ our focus is typically only on the highly relevant regions (i.e., the sorting of the h_p values on the non-relevant regions is not important). The quantity of interest in this case is the area over the MoRF perturbation curve (AOPC):

$$\text{AOPC} = \frac{1}{L+1}\Big\langle \sum_{k=0}^{L} f(\boldsymbol{x}^{(0)}_{\text{MoRF}}) - f(\boldsymbol{x}^{(k)}_{\text{MoRF}}) \Big\rangle_{p(\boldsymbol{x})} \tag{12}$$

where $\langle\cdot\rangle_{p(\boldsymbol{x})}$ denotes the average over all images in the data set. An ordering of regions such that the most sensitive regions are ranked first implies a steep decrease of the graph of MoRF, and thus a larger AOPC.

IV. Experimental Results

In this section we use the proposed heatmap evaluation procedure to compare heatmaps computed with the LRP algorithm [28], the deconvolution approach [19] and the sensitivity-based method [26] (Section IV-B) to a random order baseline. Exemplary heatmaps produced with these algorithms are displayed and discussed in Section IV-C. At the end of this section we briefly investigate the correlation between heatmap quality and network performance.

A. Setup

We demonstrate the results on a classifier for the MIT Places data set [31] provided by the authors of this data set and the Caffe reference model [40] for ImageNet. We kept the classifiers unchanged. They are both near state of the art convolutional neural networks and consist of layers of convolution, ReLU and max-pooling neurons. Both classifiers share the same architecture proposed in [41], namely

$$\begin{aligned} &\texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{Local Norm} \rightarrow \texttt{Max-Pool} \rightarrow \\ &\texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{Local Norm} \rightarrow \texttt{Max-Pool} \rightarrow \\ &\texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \\ &\texttt{Max-Pool} \rightarrow \texttt{FC} \rightarrow \texttt{ReLU} \rightarrow \texttt{FC} \rightarrow \texttt{ReLU} \rightarrow \texttt{FC} \end{aligned}$$

The Places classifier was trained on 2,448,873 randomly select images from 205 categories and the Caffe reference model was trained on 1.2 million images of ImageNet. The MIT Places classifier is used for two testing data sets. Firstly, we compute the AOPC values over 5040 images from the MIT Places testing set. Secondly, we use AOPC averages over 5040 images from the SUN397 data set [29] as it was done in [42].

We ensured that the category labels of the images used were included in the MIT Places label set. Furthermore, for the ImageNet classifier we report results on the first 5040 images of the ILSVRC2012 data set. The heatmaps are computed for all methods for the predicted label, so that our perturbation analysis is a fully unsupervised method during test stage. Perturbation is applied to 9×9 non-overlapping regions each covering 0.157% of the image. This region size was selected, because (1) it allows to perturb a significant part of the image (15.7%) in 100 perturbation steps (for computational reasons we restricted the number of perturbation steps to 100) and (2) it approximately matches the size of convolution filters used by the trained neural network models. For completeness the appendix contains results for additional region sizes. We replace all pixels in a region by randomly sampled (from uniform distribution) values. The choice of a uniform distribution as region perturbation follows one assumption: We consider a region highly relevant if replacing the information in this region in *arbitrary* ways reduces the prediction score of the classifier; we do not want to restrict the analysis to highly specialized information removal schemes. In order to reduce the effect of randomness we repeat the process 10 times. For each ordering we perturb the first 100 regions, resulting for the 9×9 neighborhood in 15.7% of the image being exchanged. Running the experiments for 2 configurations of perturbations, each with 5040 images, takes roughly 36 hours on a workstation with 20 (10×2) Xeon HT-Cores. Given the above running time and the large number of configurations reported here, we considered the choice of 5040 images as sample size a good compromise between the representativity of our result and computing time.

B. Quantitative Comparison of Heatmapping Methods

We quantitatively compare the quality of heatmaps generated by the three algorithms described in Section II. As a baseline we also compute the AOPC curves for random heatmaps (i.e., random ordering $\mathcal{O}$). Figure 4 displays the AOPC values as function of the perturbation steps (i.e., L) relative to the random baseline.

From the figure one can see that heatmaps computed by LRP have the largest AOPC values, i.e., they better identify the relevant (wrt the classification tasks) pixels in the image than heatmaps produced with sensitivity analysis or the deconvolution approach. This holds for all three data sets. The ϵ-LRP formula (see Eq. 7) performs slightly better than α, β-LRP (see Eq. 8), however, we expect both LRP variants to have similar performance when optimizing for the parameters (here we use the same settings as in [28]). The deconvolution method performs as closest competitor and significantly outperforms the random baseline. Since LRP distinguishes between positive and negative evidence and normalizes the scores properly, it provides less noisy heatmaps than the deconvolution approach (see Section IV-C) which results in better quantitative performance. As stated above sensitivity analysis targets a slightly different problem and thus provides quantitatively and qualitatively suboptimal explanations of the classifier's decision. Sensitivity provides local explanations, but may fail to capture the global features of a particular class. In this context see also the works of Szegedy [21], Goodfellow [22] and Nguyen [43] in which changing an image as a whole by a minor perturbation leads to a flip in the class labels, and in which rainbow-colored noise images are constructed with high classification accuracy.

The heatmaps computed on the ILSVRC2012 data set are qualitatively better (according to our AOPC measure) than heatmaps computed on the other two data sets. One reason for this is that the ILSVRC2012 images contain more objects and less cluttered scenes than images from the SUN397 and MIT Places data sets, i.e., it is easier (also for humans) to capture the relevant parts of the image. Also the AOPC difference between the random baseline and the other heatmapping methods is much smaller for the latter two data sets than for ILSVRC2012, because cluttered scenes contain evidence almost everywhere in the image whereas the background is less important for object categories.

An interesting phenomenon is the performance difference of sensitivity heatmaps computed on SUN397 and MIT Places data sets, in the former case the AOPC curve of sensitivity heatmaps is even below the curve computed with random ranking of regions, whereas for the latter data set the sensitivity heatmaps are (at least initially) clearly better. Note that in both cases the same classifier [31], trained on the MIT Places data, was used. The difference between these data sets is that SUN397 images lie outside the data manifold (i.e., images of MIT Places used to train the classifier), so that partial derivatives need to explain local variations of the classification function $f(x)$ in an area in image space where f has not been trained properly. This effect is not so strong for the MIT Places test data, as they are closer to the images used to train the classifier. Since both LRP and deconvolution provide global explanations, they are less affected by this off-manifold testing.

We performed above evaluation also for both Caffe networks in training phase, in which the dropout layers were active. The results are qualitatively the same to the ones shown above. The LRP algorithm, which was explicitly designed to explain the classifier's decision, performs significantly better than the other heatmapping approaches. We would like to stress that LRP does not artificially benefit from the way we evaluate heatmaps as region perturbation is based on a assumption (good heatmaps should rank pixels according to relevance wrt to classification) which is independent of the relevance conservation principle that is used in LRP. Note that LRP was originally designed for binary classifiers in which $f(x) = 0$ denotes maximal uncertainty about prediction. The classifiers used here were trained with a different multiclass objective, namely that it suffices for the correct class to have the highest score. One can expect that in such a setup the state of maximal uncertainty is given by a positive value rather than $f(x) = 0$. In that sense the setup here slightly disfavours LRP. However we refrained from retraining because it was important for us, firstly, to use classifiers provided by other researchers in an unmodified manner, and, secondly, to evaluate the robustness of LRP when applied in the popular multi-class setup.

In addition to perturbation experiments heatmaps can also be evaluated wrt to their complexity (i.e., sparsity and random-

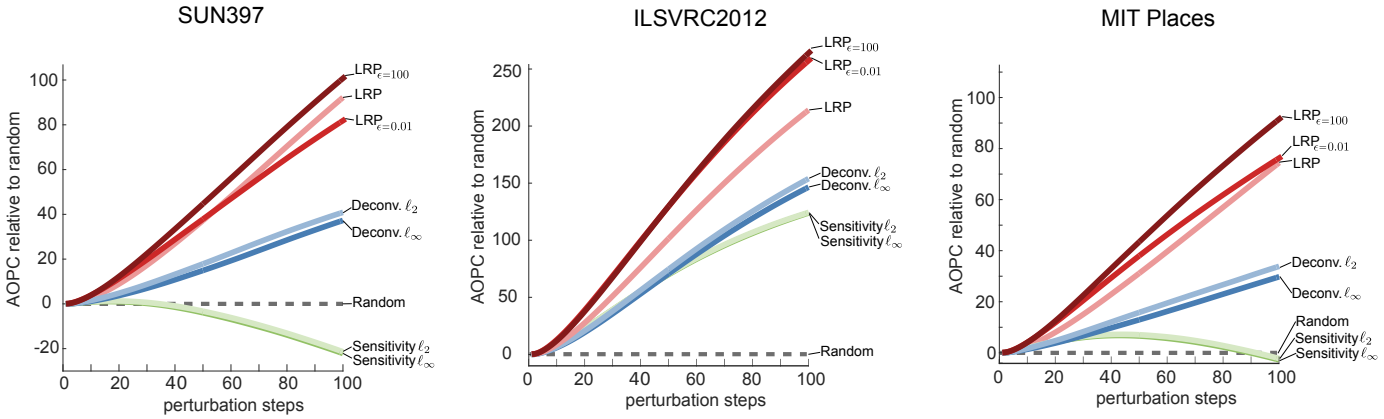

Fig. 4. Comparison of the three heatmapping methods relative to the random baseline. The LRP algorithms have largest AOPC values, i.e., best explain the classifier's decision, for all three data sets.

ness of the explanations). Good heatmaps highlight the relevant regions and not more, whereas suboptimal heatmaps may contain lots of irrelevant information and noise. In this sense good heatmaps should be better compressible than noisy ones. The scatter plots in Figure 5 show that for almost all images of the three data sets the LRP heatmap png files are smaller (i.e., less complex) than the corresponding deconvolution and sensitivity files (same holds for jpeg files). Additionally, we report results obtained using another complexity measure, namely MATLAB's `entropy` function. Also according to this measure LRP heatmaps are less complex (see boxplots in Figure 5) than heatmaps computed with the sensitivity and deconvolution methods.

C. Qualitative Comparison of Heatmapping Methods

In Figure 6 the heatmaps of the first 8 images of each data set are visualized. Red color indicates large scores, blue color indicates negative scores (only for LRP). The quantitative result presented above are in line with the subjective impressions. The sensitivity and deconvolution heatmaps are noisier and less sparse than the heatmaps computed with the LRP algorithm, reflecting the results obtained in Section IV-B. For SUN 397 and MIT Places the sensitivity heatmaps are close to random, whereas both LRP and deconvolution highlight some structural elements in the scene (e.g., mountain shape for the class "volcano" or the arch shape for the class "abbey"). We remark that this bad performance of sensitivity heatmaps does not contradict results like [21], [22]. In the former works, an image gets modified as a whole, while in this work we are considering the quality of selecting local regions and ordering them. Furthermore gradients require to move in a very particular direction for reducing the prediction while we are looking for most relevant regions in the sense that changing them in any kind (i.e., randomly) will likely destroy the prediction. The deconvolution and LRP algorithms capture more global (and more relevant) features than the sensitivity approach.

D. Heatmap Quality and Neural Network Performance

In the last experiment we briefly show that the quality of a heatmap, as measured by AOPC, provides information about the overall DNN performance. The intuitive explanation for this is that well-trained DNNs much better capture the relevant structures in an image, thus produce more meaningful heatmaps than poorly trained networks which rather rely on global image statistics. Thus, by evaluating the quality of a heatmap using the proposed procedure we can potentially assess the network performance, at least for classifiers that were based on the same network topology. Note that this procedure is based on perturbation of the input of the classifier with the highest predicted score. Thus this evaluation method is purely unsupervised and does not require labels of the testing images. Figure 7 depicts the AOPC values and the performance for different training iterations of a DNN for the CIFAR-10 data set [32]. We did not perform these experiments on a larger data set since the effect can still be observed nicely in this modest data size. The correlation between both curves indicates that heatmaps contain information which can potentially be used to judge the quality of the network. This paper did not indent to profoundly investigate the relation between network performance and heatmap quality, this is a topic for future research.

V. Conclusion

Research in DNN has been traditionally focusing on improving the quality, algorithmics or the speed of a neural network model. We have studied an orthogonal research direction in our manuscript, namely, we have contributed to furthering the understanding and transparency of the decision making implemented by a trained DNN: For this we have focused on the heatmap concept that, e.g., in a computer vision application, is able to attribute the contribution of individual pixels to the DNN inference result for a novel data sample. While heatmaps allow a better intuition about what has been learned by the network, we tackled the so far open problem of quantifying the quality of a heatmap. In this manner different heatmap algorithms can be compared quantitatively and their properties and limits can be related. We proposed a region perturbation strategy that is based on the idea that flipping the most salient pixels first should lead to high performance decay. A large AOPC value as a function of perturbation steps was shown to provide a good measure

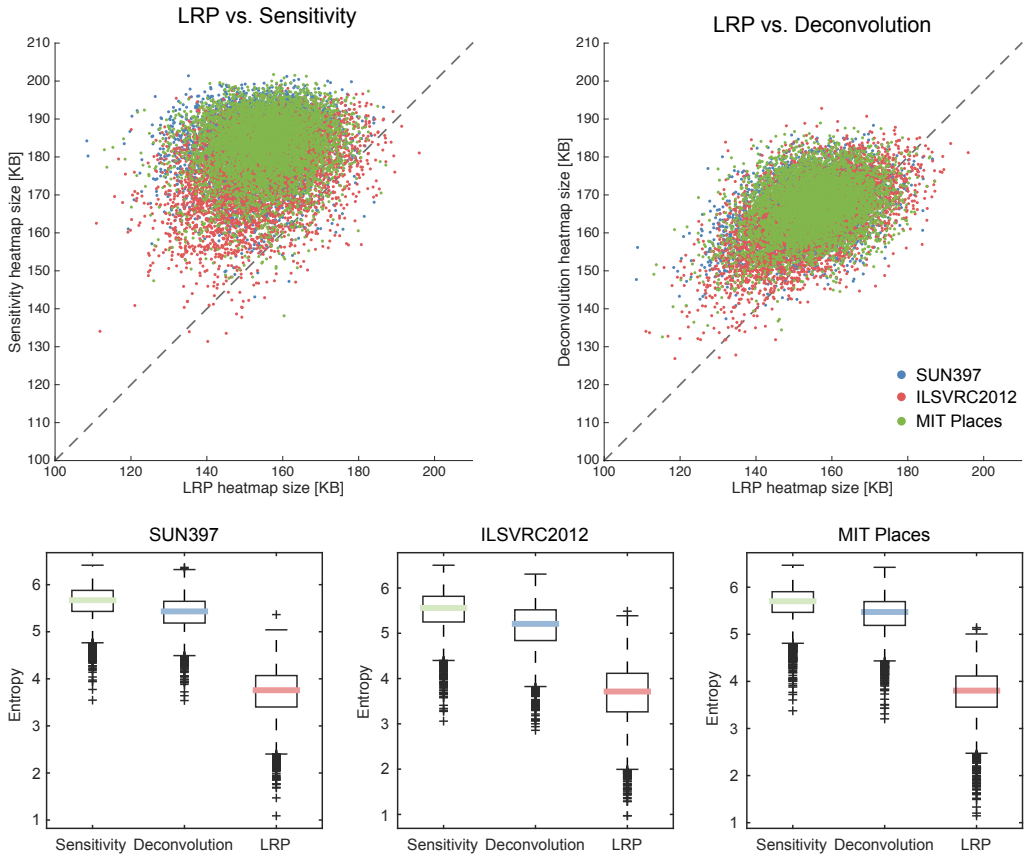

Fig. 5. Comparison of heatmap complexity, measured in terms of file size (top) and image entropy (bottom).

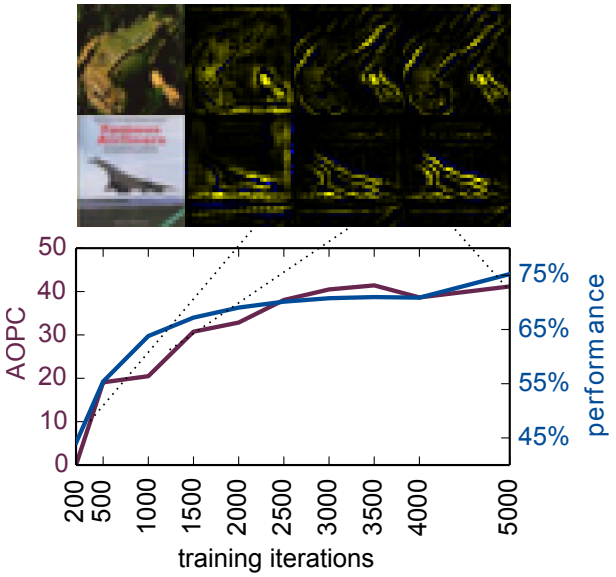

Fig. 7. Evaluation of network performance by using AOPC on CIFAR-10.

for a very informative heatmap. We also showed quantitatively and qualitatively that sensitivity maps and heatmaps computed with the deconvolution algorithm are much noisier than heatmaps computed with the LRP method, thus are less suitable for identifying the most important regions wrt the classification task. Above all we provided first evidence that heatmaps may be useful for assessment of neural network performance. Bringing this idea into practical application will be a topic of future research. Concluding, we have provided the basis for an accurate quantification of heatmap quality.

Note that a good heatmap can not only be used for better understanding of DNNs but also for a priorization of image regions. Thus, regions of an individual image with high heatmap values could be subjected to more detailed analysis. This could in the future allow highly time efficient processing of the data only *where it matters*.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. in NIPS 25*, 2012, pp. 1106–1114.

[2] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Adv. in NIPS*, 2012, pp. 2852–2860.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[4] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. of CVPR*. IEEE, 2012, pp. 3642–3649.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *JMLR*, vol. 12, pp. 2493–2537, 2011.

[6] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. of EMNLP*, 2013, pp. 1631–1642.

[7] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," in *Proc. of ICML*, 2010, pp. 495–502.

[8] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. of CVPR*, 2011, pp. 3361–3368.

Fig. 6. Qualitative comparison of the three heatmapping methods for the first 8 images of the SUN397, ILSVRC2012 and MIT Places dataset. Red color indicates large scores (see Eq. 3 for Sensitivity, Eq. 5 for Deconv. and Eq. 8 for LRP), blue color indicates negative scores (only for LRP). The heatmaps computed with the LRP algorithm focus on the relevant features of the object class (e.g., face of the dog or volcano shape), whereas the sensitivity and deconvolution heatmaps are nosier and less focused. This qualitative observations are in line with the quantitative results in Fig. 4 and 5.

[9] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Machine learning of molecular electronic properties in chemical compound space," *New Journal of Physics*, vol. 15, no. 9, p. 095003, 2013.

[10] G. Montavon, G. B. Orr, and K.-R. Müller, Eds., *Neural Networks: Tricks of the Trade, Reloaded*, 2nd ed., ser. Lecture Notes in Computer Science (LNCS). Springer, 2012, vol. 7700.

[11] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *JMLR*, vol. 3, pp. 1157–1182, 2003.

[13] M. L. Braun, J. Buhmann, and K.-R. Müller, "On relevant dimensions in kernel feature spaces," *JMLR*, vol. 9, pp. 1875–1908, 2008.

[14] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *JMLR*, vol. 11, pp. 1803–1831, 2010.

[15] K. Hansen, D. Baehrens, T. Schroeter, M. Rupp, and K.-R. Müller, "Visual interpretation of kernel-based prediction models," *Molecular Informatics*, vol. 30, no. 9, pp. 817–826, 2011.

[16] G. Montavon, M. L. Braun, T. Krueger, and K.-R. Müller, "Analyzing local structure in kernel-based learning: Explanation, complexity and reliability assessment," *Signal Processing Magazine, IEEE*, vol. 30, no. 4, pp. 62–74, 2013.

[17] D. Erhan, A. Courville, and Y. Bengio, "Understanding representations learned in deep architectures," *Universite de Montreal/DIRO, Montreal, QC, Canada, Tech. Rep*, vol. 1355, 2010.

[18] G. Montavon, M. Braun, and K.-R. Müller, "Kernel analysis of deep networks," *JMLR*, vol. 12, pp. 2563–2581, 2011.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014, pp. 818–833.

[20] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," *CoRR*, vol. abs/1412.0035, 2014.

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014.

[23] J. Yosinski, J. Clune, A. M. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *CoRR*, vol. abs/1506.06579, 2015.

[24] A. Dosovitskiy and T. Brox, "Inverting convolutional networks with convolutional networks," *CoRR*, vol. abs/1506.02753, 2015.

[25] P. M. Rasmussen, T. Schmah, K. H. Madsen, T. E. Lund, G. Yourganov, S. C. Strother, and L. K. Hansen, "Visualization of nonlinear classification models in neuroimaging - signed sensitivity maps," in *BIOSIGNALS*, 2012, pp. 254–263.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *ICLR Workshop*, 2014.

[27] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, "Analyzing classifiers: Fisher vectors and deep neural networks," in *Proc. IEEE CVPR*, 2016, pp. 2912–2920.

[28] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, p. e0130140, 2015.

[29] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. on CVPR*, 2010, pp. 3485–3492.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, pp. 1–42, 2015.

[31] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Adv. in NIPS*, 2014, pp. 487–495.

[32] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, University of Toronto, 2009.

[33] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[34] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, "The layer-wise relevance propagation toolbox for artificial neural networks," *JMLR*, vol. 17, no. 114, pp. 1–5, 2016.

[35] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *arXiv*, no. 1512.02479, 2015.

[36] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision research*, vol. 40, no. 10, pp. 1489–1506, 2000.

[37] D. J. Heeger, E. P. Simoncelli, and J. A. Movshon, "Computational models of cortical visual processing," *Proceedings of the National Academy of Sciences*, vol. 93, no. 2, pp. 623–627, 1996.

[38] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual review of neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001.

[39] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature reviews neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[40] Y. Jia, "Caffe: An open source convolutional architecture for fast feature embedding," http://caffe.berkeleyvision.org/, 2013.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[42] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv:1412.6856*, 2014.

[43] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," *CoRR*, vol. abs/1412.1897, 2014.

APPENDIX

We performed the perturbation experiments from Section IV-B also with region sizes smaller and larger than 9×9, namely 1×1, 3×3, 5×5 and 19×19. Figure 8 shows the results relative to the sensitivity-ℓ_2 baseline. Due to the high computational load we did not compute the random baseline and also left out the ℓ_∞ norm as it performed very similarly to ℓ_2. Since $\text{LRP}_{\epsilon=100}$ performed much better than $\text{LRP}_{\epsilon=0.01}$ in Section IV-B, we decided to leave out the latter and to include a LRP variant with an ϵ value in between 0.01 and 100. We decided to take $\epsilon = 1$.

Qualitatively the results are very similar to the one reported in Figure 4 for the 9×9 region size. In all cases the ϵ-stabilized LRP algorithm provides best explanations. Interestingly, $\epsilon = 1$ outperforms the results obtained with significantly larger or smaller stabilizing factors. Thus although stabilization is important (as discussed in Section IV-B), too large ϵ values may decrease the efficiency of LRP as it suppresses contributions from small relevances. Another observation which can be made from Figure 8 is that the deconvolution method outperforms sensitivity analysis in terms of AOPC only for large region sizes (9×9 in Figure 4 and 19×19 in Figure 8). Note that the explanations computed by deconvolution show a large number of responses that look like filter visualizations (see Figure 6). The kernel size for the used networks is 11. We argue that with a region size close to or larger than the kernel size, we are obtaining a score of the region that is close to the score of the filter as a whole. Destroying such a region has a large impact on the filter response and thus leads to a decrease of the classification score. Destroying individual pixels (or small regions) on the other hand only slightly affects the filter response and results in small AOPC values. LRP performs well for all region sizes, because it does not focus on filter responses but identifies the relevant pixels in the image. Note that these pixels often represent the shape of the object (see Figure 6), so that destroying small regions around these pixels largely destroys the object shape which leads to a fast decrease of classification score.

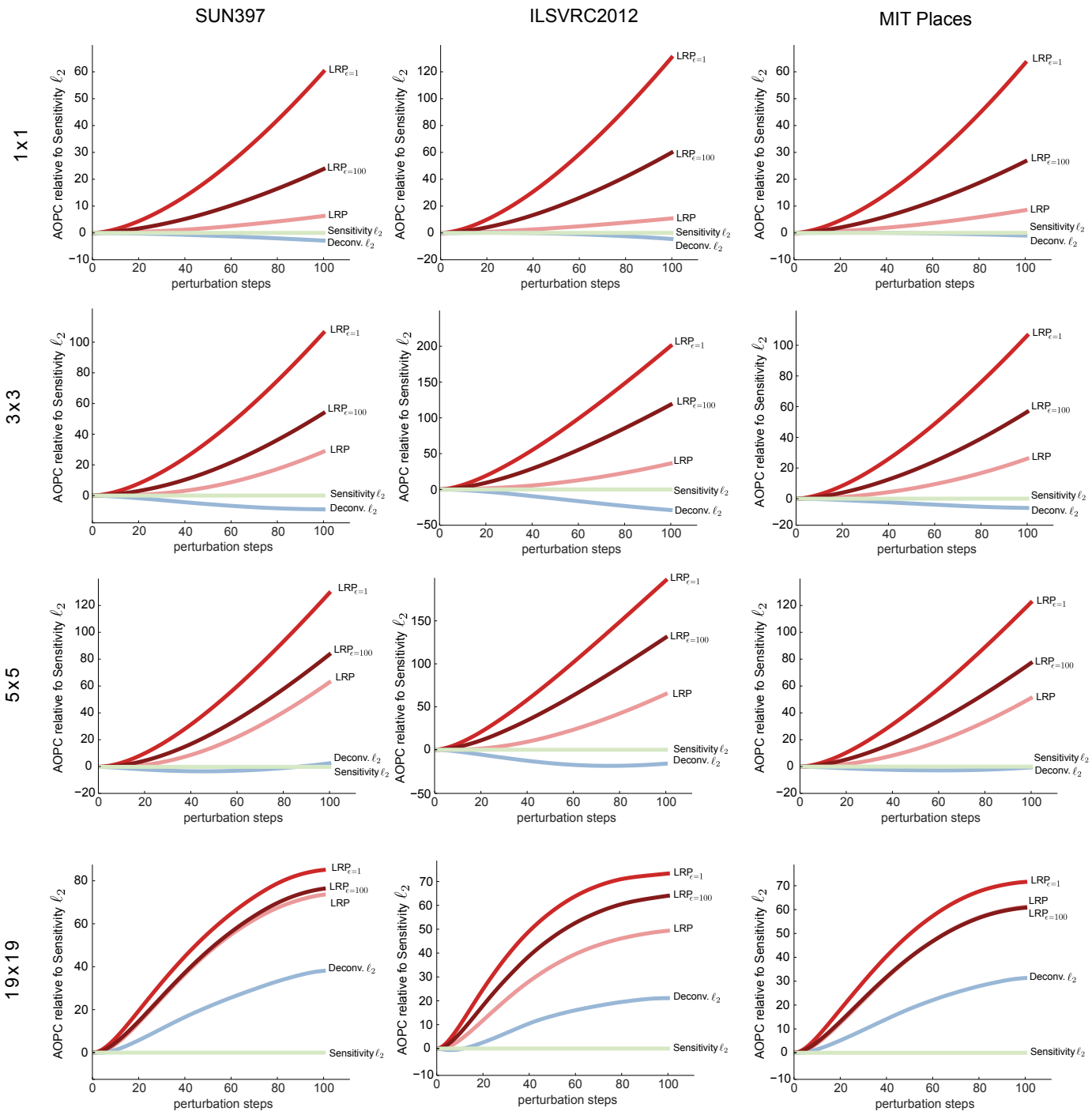

Fig. 8. Comparison of the three heatmapping methods relative to the sensitivity ℓ_2 baseline for different region sizes.